Visualization and Data Analysis with VisIt and Matlab

Elena Caraba

ECaraba@lbl.gov

LBNL/Visualization Group - LSU/Center for Computation & Technology

August 9, 2006

Overview of a Summer Internship

- Scientific Visualization
- VisIt and Matlab for combustion data on unstructured meshes and of supernova simulations
- VisIt plugins, methods and algorithms for comparative analysis
- Performance Test
- Results

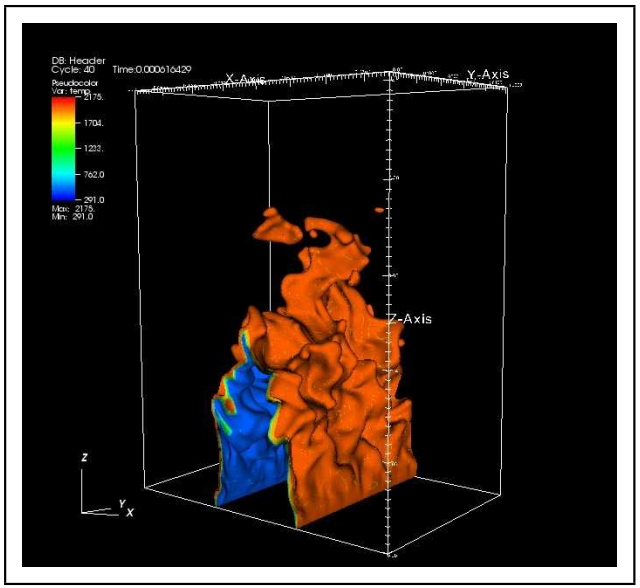
Performance Test on DaVinci

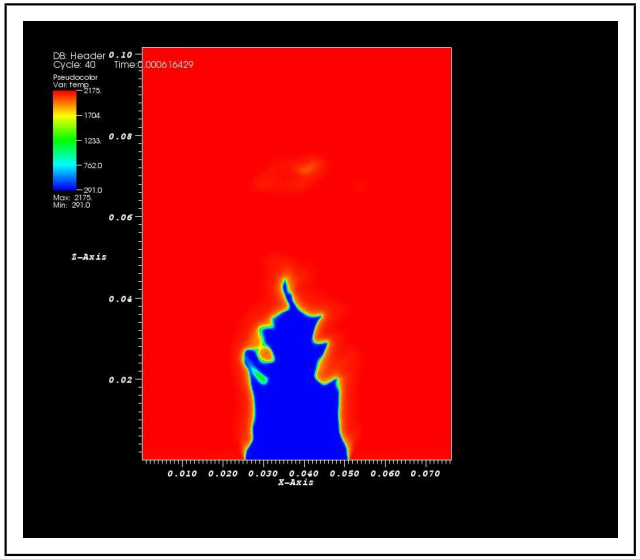
DaVinci is an SGI Altix 350 server running the SGI ProPack 4 64-bit Linux operating system, based on SUSE Linux Enterprise Server 9 (SLES9). Its main purpose is to provide visualization and data analysis capabilities to the NERSC user community. It is named after the famous Italian artist and scientist Leonardo da Vinci, due to its intended combination of visual imagery of technical information.(http://www.nersc.gov/nusers/resources/davinci/)

DaVinci Characteristics	
Processor type	Itanium 2, 1.4 GHz
Processor theoretical peak	5.6 GFlops/sec
Number of application processors	32
System theoretical peak	179.2 GFlops/sec
System physical shared memory	192 GBytes
Usable physical shared memory	180 GBytes
Global shared disk	XFS
Usable disk space	23.8 TBytes
Batch system	PBS Pro

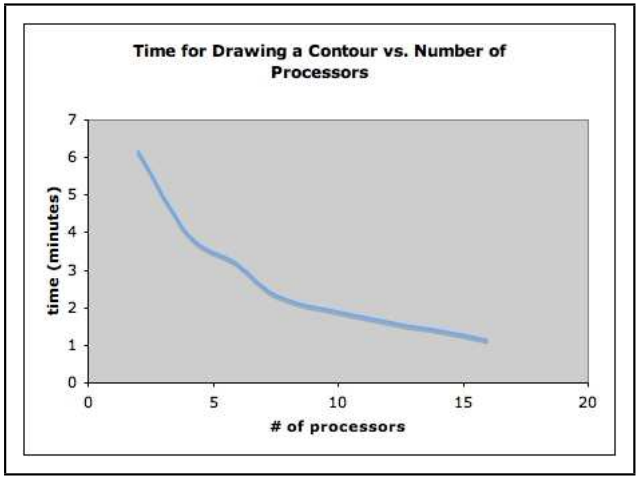
Images of Driscoll Data

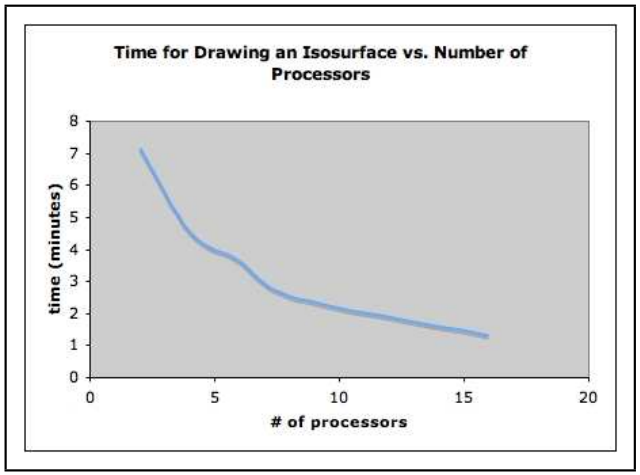
- The Driscoll data is a "medium-sized" combustion data, with a 19GB/time step.
- It has 4 levels and 6292 domains, 41 variables, and an AMR type of mesh.
- For our performance test we chose a time step equal to 0.000616429.

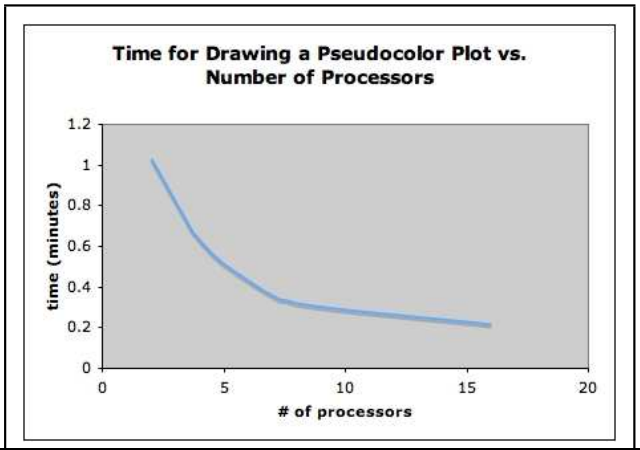


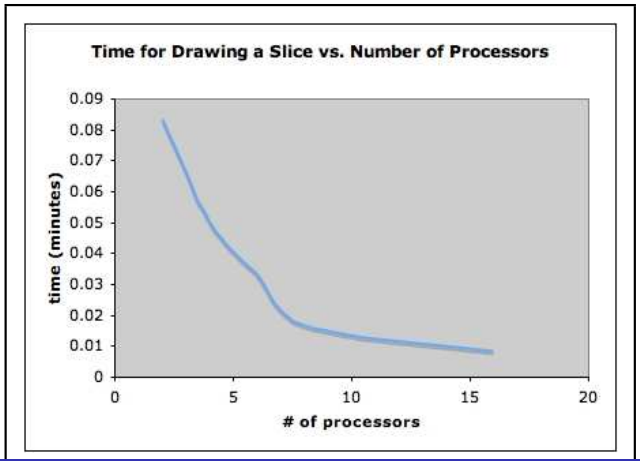


Evaluation of the Speedup Gained by Using a Parallel Engine



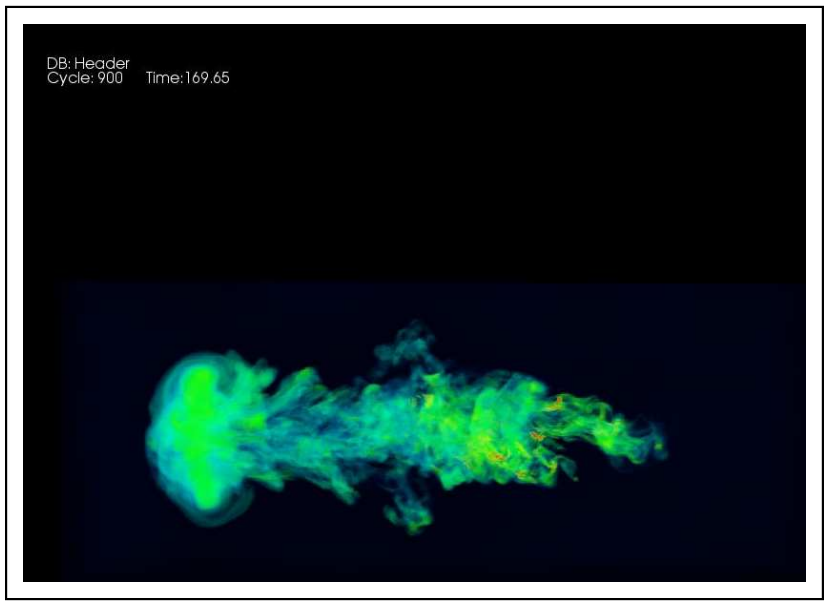






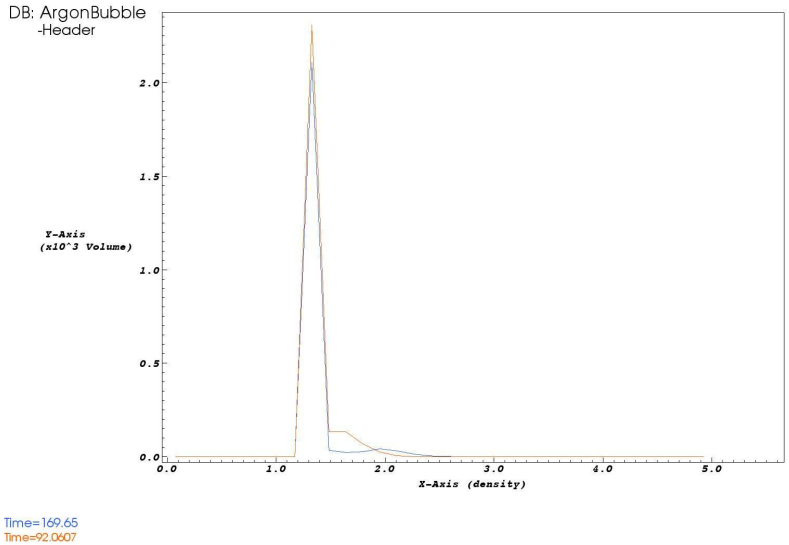
Argone Bubble Data

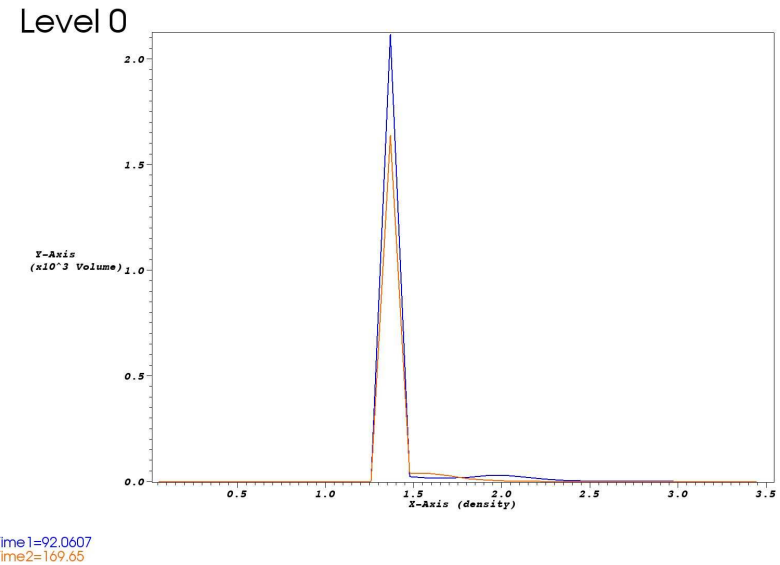
- - a small-sized data with an AMR type of mesh
- - 9 time steps between 140MB and 520MB



Comparison of Variable Distribution for Different Time Steps

- Time step= 92.0607, 292MB/ts, 3 levels, 528 patches
- Time step= 169.65, 519MB/ts, 3 levels, 738 patches
- Histograms of the density distribution over volume

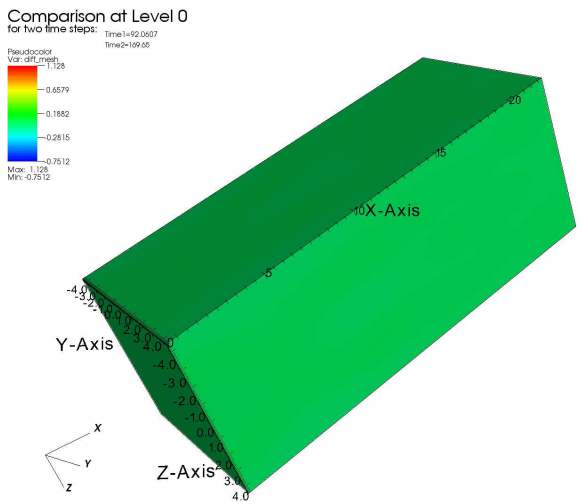


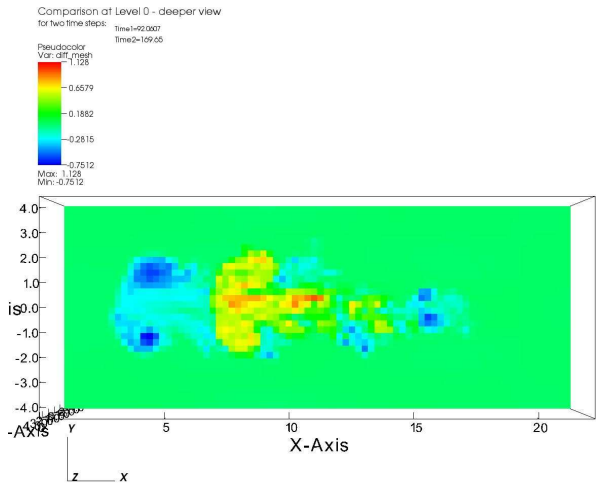


Comparison at level zero

-Comparison possible just at level zero - the only one with the same number of patches

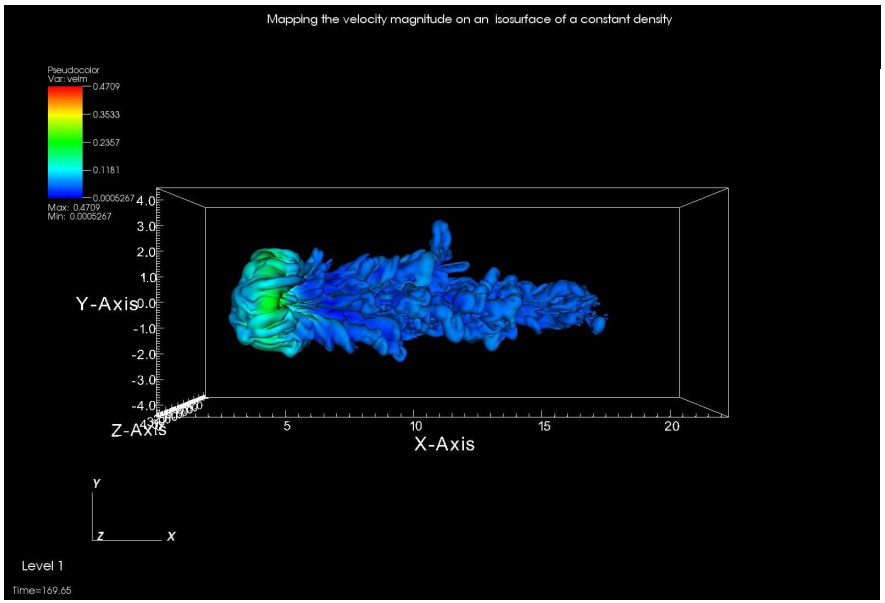
-A slice taken perpendicular to the z-axis for a Pseudocolor plot reveals the difference in density for the two time steps

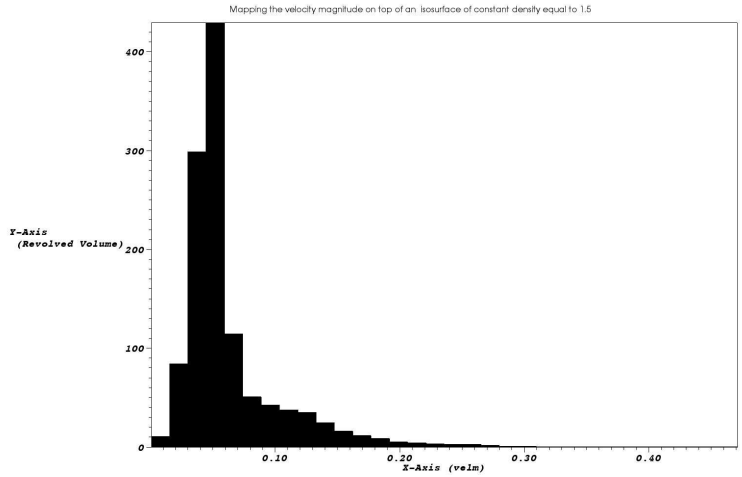




Velocity Magnitude over an Isodensity Surface

-"false-coloring" the isodensity surface according to velocity magnitude
-cells with lower velocities are located in the "tail", while those with higher velocities are in the "head"

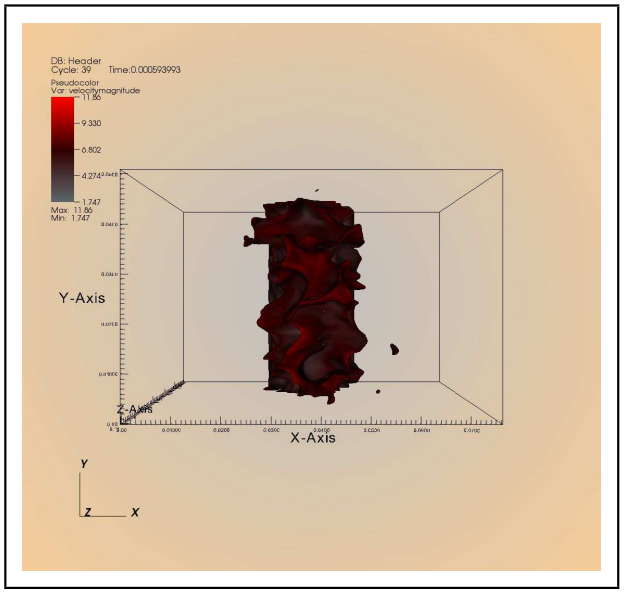


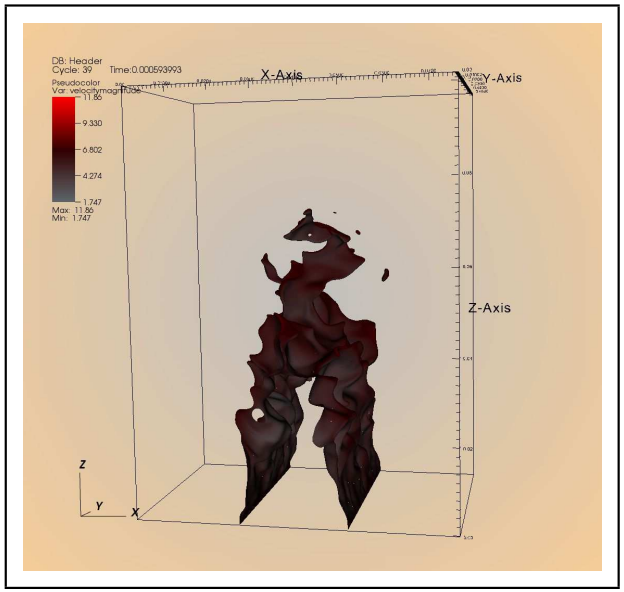


Colorizing an Isotherm with the Velocity Magnitude

-Color mapping: scalar values are mapped through a lookup table to obtain a color that would be applied to modify the apperance of the points or cells

-colorizing the isotherm 1685 with the velocity magnitude for the Driscoll data





Developing VisIt

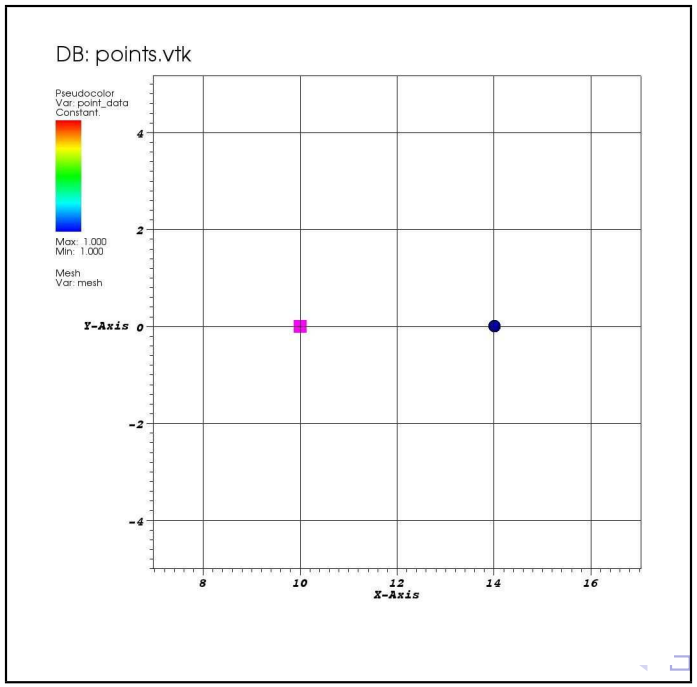
- understanding the VisIt pipeline is not an easy task
- VisIt provides an xml interface to generate C++ skeletons for operators, plots, etc.
- populating the skeleton with the appropriate VTK functions, methods, classes
- after compilation the plugin is a shared object that is picked up by VisIt
- operators can be made serial and paralallel

An operator plugin to compute mean values

- objective: make a histogram of the distribution of a certain variable
- VisIt doesn't have any operator or query for statistics
- I wrote an operator to compute the mean of a dataset
- I made this operator work in parallel

Streampoints plugin

- How can we update the renderer pipeline in VisIt?
- Plots or operators for a statical particle advection?
- plots give you access to the renderer engine, while the operators don't

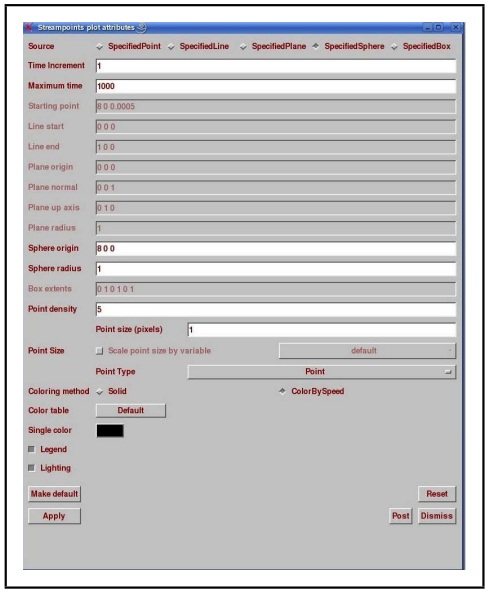


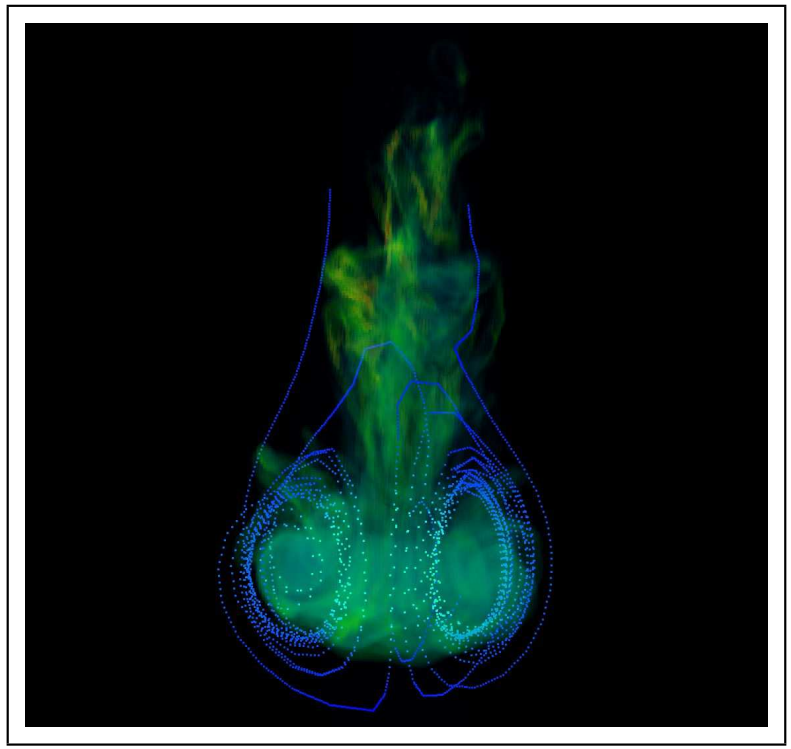
Streampoints -cont.

- - mappers transform and render the geometry
- - the vtkStreamer abstract class that includes the vtkStreamPoints and vtkStreamlines contains methods for the Runge-Kutta 4 and Integration algorithms
- - Streampoints vs. Streamlines: the former uses a time increment, the latter a step length
- - features for the streampoints: resizing the points, choosing a source plan, a method of colorizing them and the type of point

Results

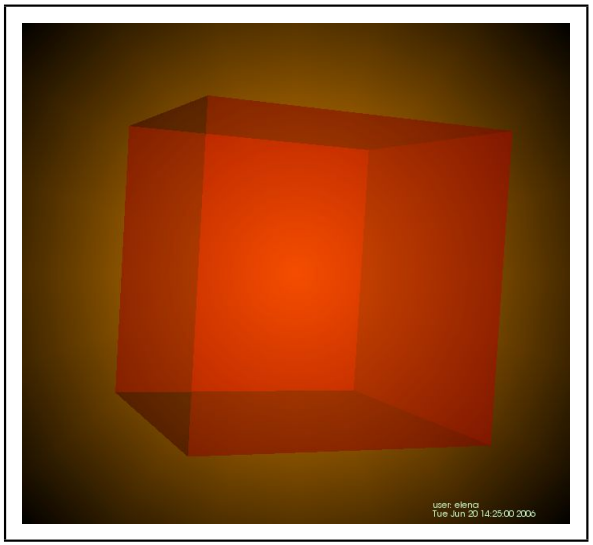
- the bigger the distance between points, the faster they would travel

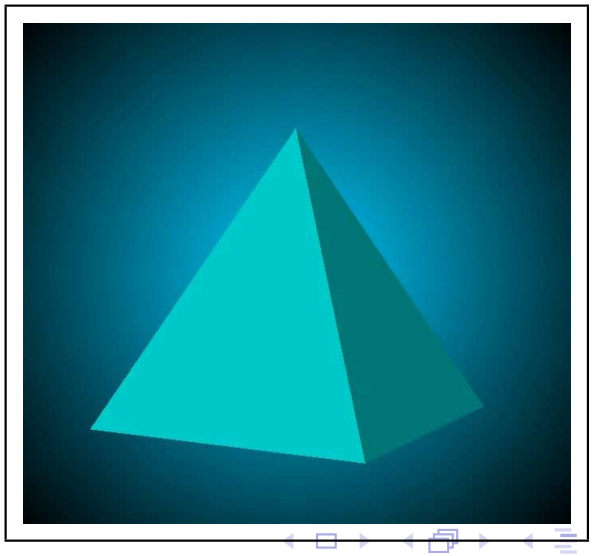




Converting *.dat to .vtk*

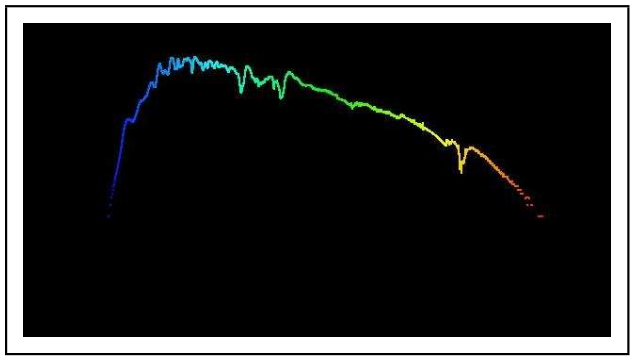
- Supernovae Spectral Templates - ASCII files with 3 columns: time, wavelength and flux
- Reading the *"VTK File Format"*, implementing the VTK example, creating our own
- Converting the data into a VTK format to be read in VisIt

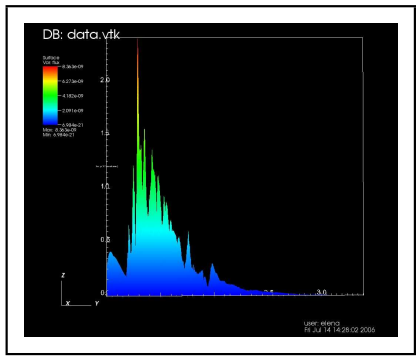




Simulated Representation of a Supernova

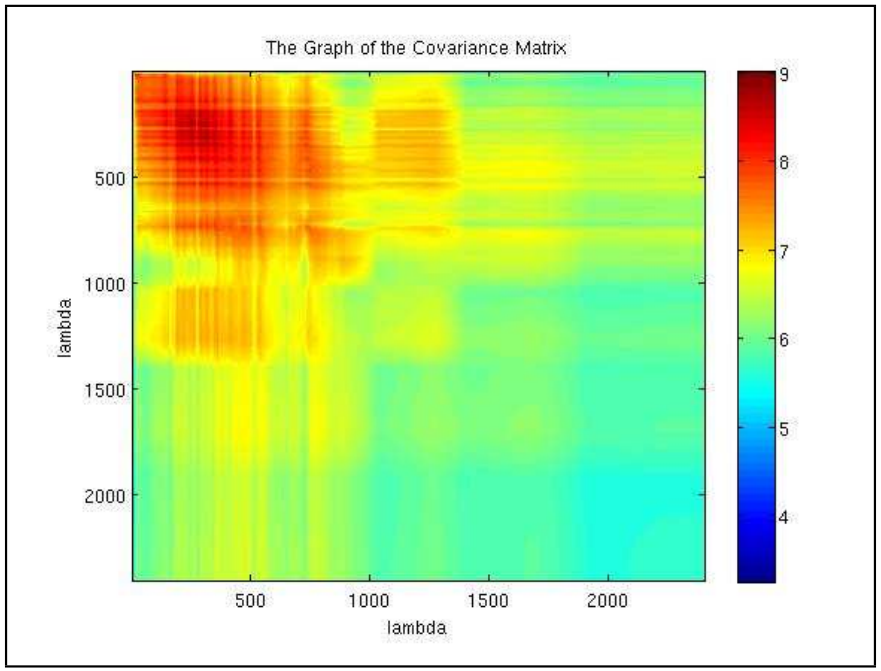
- The flux varies over time differently for each wavelength
- The higher the graph, the brighter the supernova in that particular color
- Using PCA to understand how flux varies at different wavelengths





The Covariance Matrix

- The wavelengths are more correlated at lower values
- The variance is higher at lower wavelengths



Getting the Data in Matlab

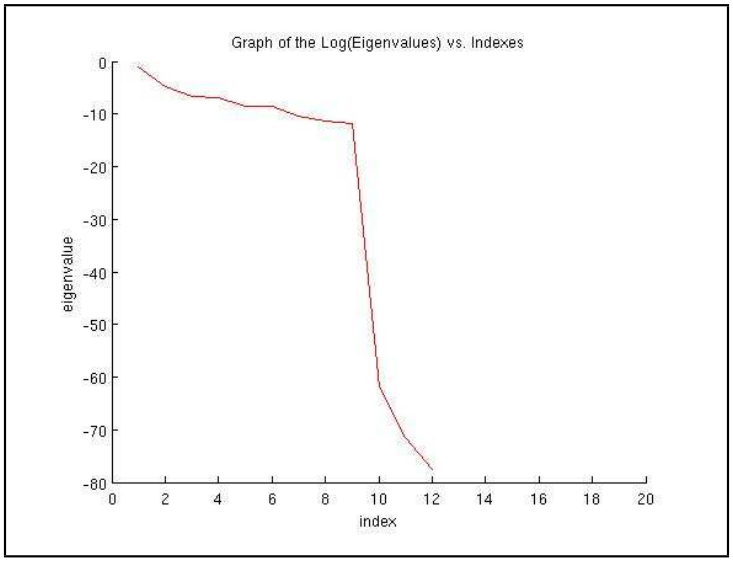
- A matrix for the entire dataset

$$\begin{pmatrix} \ldots & flux & \ldots \\ \ldots & \ldots & \ldots \\ flux & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \end{pmatrix}$$

- number of rows = number of time steps
- number of columns = number of wavelengths (2401)
- 2401 dimensions for PCA

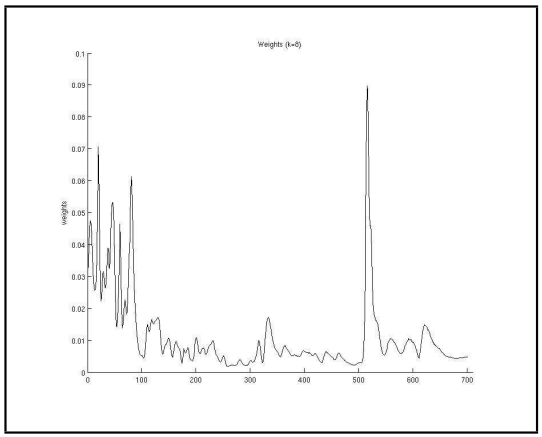
Eigenvalues, Eigenvectors

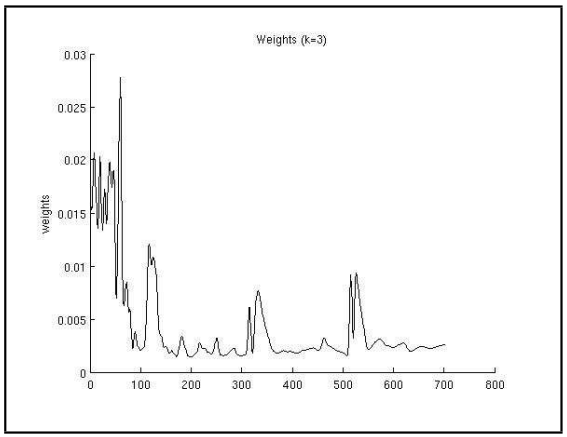
- PCA reduces the dimensionality with a minimum loss of information
- How many dimensions contain most of the information?
- I use SVD to compute and sort the eigenvalues and eigenvectors.
- The graphs of the eigenvalues give a good estimate for the number of eigenvectors we are interested in.



Weights

- Project the entire data into a 8D space defined by the first 8 eigenvectors
- Using weights to see which are the components with high and low variability





The Science Story for Data Analysis

- Using the weights with the new spectrae of supernovae, the astrophysicists can classify them by type and time
- How fast is the universe expanding?
- How old is the universe?
- +other questions related to dark energy, particle physics

What was my point?

Present my summer internship experience:

- see what VisIt can or cannot do with a data
- try to get the most out of VisIt and find its limitations
- make a data analysis
- look for features that were not previewed in the templates

Why VisIt?

- can handle large data sets
- has features for comparative visualization
- runs in distributive mode

Why Matlab?

- is easy and fast to use for matrix computations
- has built-in support for data analysis and visualization routines
- PCA, SVD, interpolation

What are the results?

- explored the features of VisIt
- created new plugins
- researched algorithms for data analysis
- ! eventually, we would be able to help the astrophysicists to compare the observational data to templates and answer some of the "big" questions